

Title	DATA BOX (情報の記憶と利用に関する理論的研究)
Author(s)	富樫, 雅文
Citation	数理解析研究所講究録 (1981), 423: 199-212
Issue Date	1981-04
URL	<a href="http://hdl.handle.net/2433/102574">http://hdl.handle.net/2433/102574</a>
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

## DATA BOX

北大 理 物理 富樫雅文

### 1. 研究活動と情報管理

Data Base 管理システム (DBMS) や情報検索システム (IRシステム) が研究者にとっても容易に利用しうる環境が出来つつあるが、これらのシステムが前提として考えている情報の性質や情報管理の形態は現実の研究活動の中での情報とその管理とくらべて隔りをもっているように思われる。

いうまでもなく、文献ファイルなどの場合のように、DBMS 等のシステムが適用可能な領域も研究活動の中の情報としてありうるが、実験ノートや各種実験データの整理等については依然手作業が主体であって、計算機を用いてのこれらの情報の管理は未だ定着していない。個人の研究活動の中では、一般に定型化しにくい情報ととりあつかわることが多く、その処理もまた非定型のものが多い。とりあつかわれる情報のまたる特徴は、それが発生段階の情報や収集直後の未整理な状態にあるという点である。また情報処理という面からは

ルーチン的というよりは研究者の思考に沿った処理であり、  
任意の時点で任意の方法をとって行なわれる。

研究活動の中での非定形の情報と非定型の情報処理は、  
研究が進み、知識が整理されるに従って定型化へと進み、組  
織化された情報管理へ移行しうる部分を生じてくる。そこか  
ら先の過程についてはDBMS的アプローチが有効になるも  
のと思われる。問題は、未整理かつ未評価の情報からより高  
度な情報組織化への漸近的移行をどのように保障・支援する  
かという点にある。さらに未整理の情報に対してそれをその  
まま入力し計算機として整理作業を進められる、いわば、Data  
Baseへ至る道の出発点を確保するシステムを作ることが重要  
な課題である。

このシステムのひとつとしてここに提案する Data Box  
は、計算機上で実現される情報の整理箱であって、受け入れ  
る情報の形式に細かな基準を設けず、また収容した情報を自  
由に閲覧し、さらに必要に応じてそれらを編集することもめ  
ざったシステムである。

## 2. Data Box

Data Boxは文字通り Box (箱) であって、現実世界での  
箱がもつ機能、すなわち、収容・尾開・箱ごとの操作、内容

の一部の出し入れ等がなければならぬ。箱は、現実世界で  
 そうであるように、単に“もの”が位置を占めるための広がり  
 ——空間——だけでなく、“もの”へのアクセスと操作を  
 保障する仕組み——機構——でもある。Data Box はこの意  
 味で情報を入れる箱であり、空間と機構のふたつの性質をも  
 つ。この箱は計算機上で論理的に実現されたものであって、  
 そこでは物理的制約を超えて箱を作ることができる。すなわ  
 ち、箱の生成・消滅・融合・分割や入れ子構造などのほかに  
 互いに重なり合、て内容の一部を共有するといった非物理的  
 な箱も実現可能となる。

Data Box の主要機能は、情報の保管・閲覧・組織化で  
 ある：

#### (1) 情報の保管

箱の基本機能として、入力された情報を任意のグ  
 ループごとに箱に入れ、箱ごとのアクセスや内容  
 の提示を可能にする。

#### (2) 情報の閲覧

以下の視点移動によって内容を見る。

- a) Pan 水平移動：同一水準の情報を列挙す  
 る。
- b) Zoom-in 接近：より低い水準の情報をと

り出してやる。

c) zoom-out 後退 : より高い水準の情報をとり出してやる。

ここで "水準" とは、箱が入れ子になっている状態のときに、より内側の箱を低水準、より外側の箱を高水準とする。

すなわち、zoom-in は入れ子の箱を内側へ次々と入っていく操作であり、zoom-out は逆に、内側から出発して、着目する箱を包含している外側の箱へとたどっていく操作である。

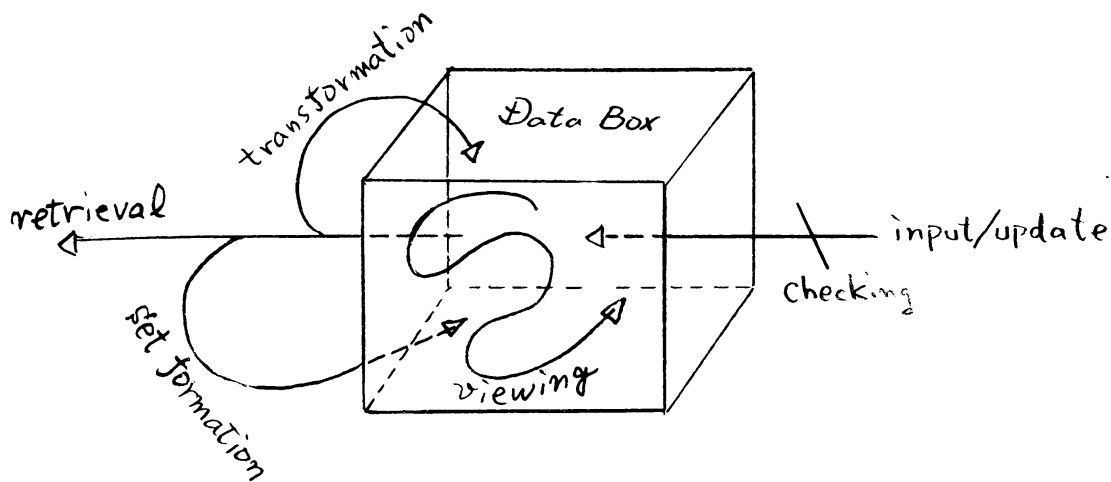
### (3) 情報の組織化

ここでいう組織化とは、情報の新たなグループ化とグループへの名前づけによって達成できる程度のもものを指す。

この組織化の過程は、閲覧によって人間が得た認識を新しいグループの定義という形で表現して Data Box へ記録する帰還過程であって、認識の固定化とそれによる情報構造の追加が行なわれる。

図1は Data Box における情報の流れを示す。

図1. Data Box における情報の流れ



### 3. Data Box と Data Base

Data Box はどのような意味において Data Base ではないのか？

ともに現実世界からある filter を通して情報を採取してはいるが、Data Base においてはその filter が既知であり、かつ現実世界もまた既知であると仮定している。一方、Data Box の場合、現実世界とは研究の対象たる未知の世界であり、そこから実験・観察という作業によって得られた情報が蓄えられることになる。実験や観察は一種の filter として認められるが、Data Base における filter のように世界に対する理解の枠組といったものではなく、情報変換枠組とみる方が近い。

Data Base における filter とはすなわち schema をさす。世界がどのように構成されているか、構成要素は互いにどのように関連しているか、何があり得て何があり得ないか、等の理解を schema は内包しているともみることが出来る。研究過程を考えた観点からすると、この schema は研究の goal を意味している：すなわち、対象とする物や現象の構造・法則に対する情報は、研究によって明らかにされるべきものである。前提ではない。

世界の理解を得るということが Data Base においては、その出発点であるのに対して、Data Box では goal として認識されるという。この差異は両者の本質的なちがいをあらわす（同時に、両者の相補的性格も暗示しているようにみえる）。

#### 4. Data Box の設計

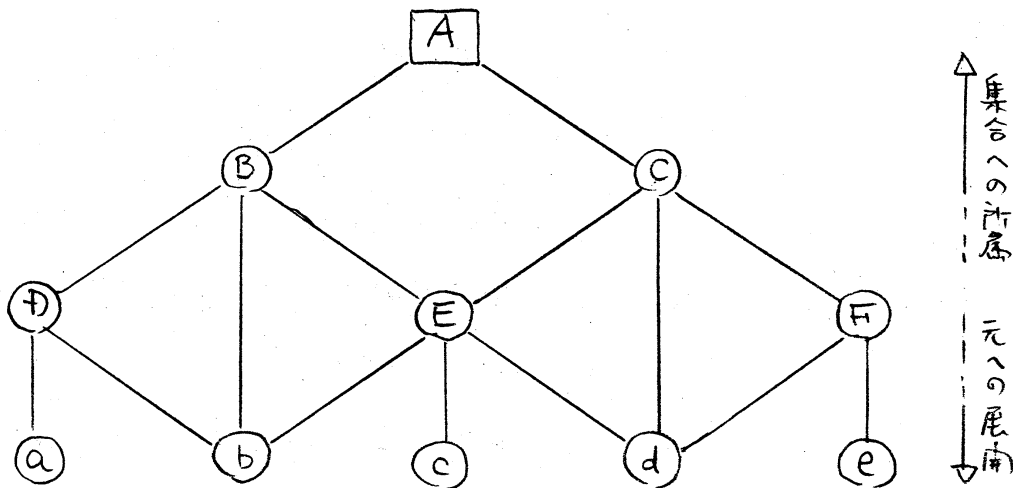
##### (1) 基本情報構造

Data Box は計数材料としては "hyper set" として実現される。以下のようにして構成される構造をここでは "hyper set" と呼ぶ：

あるひとつの基底集合を出発点としてそこから部分集合族をとり、そのあつての元、すなわち各部分集

合に対応する元、とすべての集合のすべての元を合わせてひとつの集合とみなし、これにまた同じ手順をくり返し適用して得られる集合を“hyper set”とする。  
hyper set では、集合をひとつの元としてとらえる際の元と集合の内の対応関係も保存している。hyper set 中に含まれる元（同時に集合でもある）を“要素”と呼ぶ。この要素は一般にまた hyper set であって、入れ子構造をしている。（図 2）

図 2 hyper set の例



$a \sim e$  は 基底集合 (primary set) の元

例えば、要素  $E$  は集合として元  $\{b, c, d\}$  をもち、  
元として集合  $B$  と集合  $C$  に含まれている



基底集合 (Primary set) は、Data Box 中の原子的情報を表わしており、そこから下位への nesting はない。primary set の元は、意味を担う最小の情報単位であり、これも分解した場合、各部分は単独では情報をもたさない。

hyper set の各要素は一般に名前をもつことから、その名前のもとに下位の直属する元の集まりを示すが、Primary set の名元の場合は、名前が内容自身であって、名前と内容という区別は存在しない。

性質の似たもの同士や互いに関連するものをひとつにまとめて単位化し、単位ごとのとりあつかいや必要な時にのみその内容の展開をすることは、日常の書類や物品の管理の際によく行われる。さらに、人間の概念形成の基本的な過程としてもあらわれる。これを“集合化”と呼ぶなら、集合化は、それ以前の状態に対するある認識をあらわし、同時に集合の内容をまとめてひとつの存在とみなしてその名前や所在のみを意識するだけである。

いうまでもなく、情報に構造を入れる手法としては、この hyper set に限らない。Data Box が hyper set をその基本構造としてとる理由は、集合—元という単純な関係を基礎としてその積み重ねで構造化するという“構成的”なアプローチがとれるということにある。Data Box では informatical-

semantics については無色透明であることを道具という立場から指向する。

## (2) 基本情報操作

hyper set に対する操作が Data Box での基本操作となる。集合演算を除いて主要なものは以下の通りである。

### a) creation of set

元を指定して集合を生成する

### b) deletion of set

集合の消去; 消去はその集合の直属の元には及ばない

### c) explosion of set to elements

集合の元を列挙する

### c) implosion of element into sets

要素を元とみた時の直属する集合を列挙

### d) enclosure of elements

集合中のいくつかの元をひとつの集合にまとめ要素化する。この要素が代表して元となる。

### e) exposure of elements

集合中の元がまた集合にあるとき、もとの元も下位の集合のすべての元とおきかえる。

f) inclusion of element into set

集合に新たに元を加える

g) exclusion of element from set

集合から元を排除する

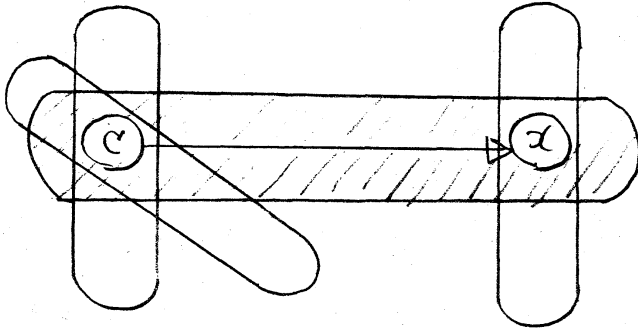
h) separation of set into sets

集合を二つまたはそれ以上の集合に直和的に分解する

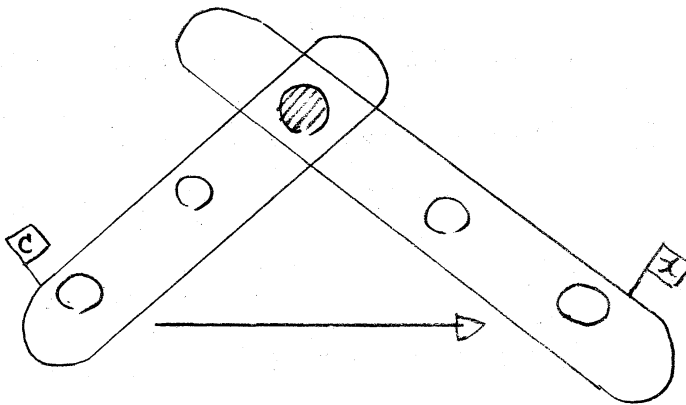
これらの操作は必ずしも互いに独立ではなく、あるものは他の操作の複合として実施することからできる。これらの中で、explosion と implosion は Data Box に特徴的な操作である。これによって単一のものと1つの集合から元へ (zoom-in), また元から集合へ (zoom-out) の視点の移動が行われる。hyper set の深さ方向へのこの視点移動のほかに, explosion, implosion 及び要素の選択をくり返し適用することによって、集合から共通元を介して別の集合へ, また、元から共通集合を介して別の元へ渡っていく, いわゆる structure-traverse も可能である。これは視点の水平移動のひとつである。(図3)

集合の生成操作は、データの入力時以外にも行われる。それは、検査負荷によってひとつの新しい集合を得た場合である。このとき、見いだされた集合に名前をつけて保存す

図3 structure traverse



$$x = \text{projection}(a) \text{ explosion}(\text{projection}(b) \text{ implosion}(c))$$



$$x = \text{projection}(a) \text{ implosion}(\text{projection}(b) \text{ explosion}(c))$$

(projection( ) は ある基準での要素の選択を表わす)

ることが集合の生成にあたる。このほか、Data Box へ  
任意の選択によって要素もとりに出してそれを一つの箱に入  
れるという形の、収集型の集合形成と、いくつかの箱を用意  
しておいて、それぞれへ要素を分配する分配型の集合形成があ  
る。いずれも資料整理法として日常、物理的に行われていた

手法の simulation である。

## 5. 情報表現

### (1) 情報要素

hyper set の最深部, すなわち primary set の各元はもはや集合を使った構造的情報としては表現できない性質の情報を保持している。この要素としては、次の 2 種類の形式の

- ・ type name と instance の組
- ・ symbolic code

type name と instance は名々ひとりの code で表現するものとする。また symbolic code は単一の code である。いずれの場合についても、その意味するものについては、Data Box の枠内では与えられず、利用する人間の側で code と意味の対応づけを行なわなければならない。

hyper set の中への random entry の方法は、この code もしくは code の組を指定するが、あるいは直接に、集合名を指定することによって行なわれ、その後は、implosion explosion 等の基本操作によって hyper set の中を移動する。

### (2) 表

表のもつ情報表現能力は日常生活においてもよく認識されており、表現しようとする情報に向らかの一様性が認めら

れる時、この一様性を利用して各要素も空間的にうまく配置することによって一覽性が高められ、内容の迅速な理解を促す効果が見られる。すなわち、情報の空間的表示の方法として人間とのよい interface をもっているといえる。したがって計算機の外部にいる人間との情報のやりとりに表を媒介することは、交換される情報に一様性が存在する場合には望ましいことである。しかし、計算機内では、表を物理的に空間配置する必要はない。

表は、Data Box の hyper set の中で一つの集合として保持される。

表の拡張として tensor 型のデータの並びすなわち  $n$  次元配列を考える。この各要素は  $n$  個の suffix によって指定される。そこで各要素自身に  $n$  個の suffix の並びを付加してこの suffix list と値の組をあらためて一つの要素としてそれを集め集合を作る。この、いわば冗長化の処理によって表を集合の枠組みの中に入れ、presentation の段階で解釈によって空間配置を作り出す。

2次元の場合、すなわち表の場合、各要素に対して、column-name と row-name の組を一つの type name とみなせば、前述の type-instance の組と同等にあつかうことができる。

表の本質をあくまで空間配置の一方法とみるならば、それは、人間との interface であり、データ基上での論理的表現とは独立であると考ええる。

## 6. まとめ

学術情報組織化の初期過程を担うシステムとして、Data Box を提案した。Data Box は nesting を有する hyper set を基本情報構造とし、hyper set の編集を基本情報操作としてもち、データの入力・検索のほか、hyper set 中に新たな集合を付加していく帰還過程によって情報整理を行なうシステムである。

帰還は、多重の元—集合関係という構造と、その下位の母集合にある code 群の両者によって担われる。hyper set の中へは、各要素への random entry とその後の視点移動によって接点する。

Data Box は、学術情報の管理に因りて Data Base と対向に相補的な関係をもつものとして位置づけられるものと考ええる。